Insper

Workshop

Limpeza e organização de bases de dados

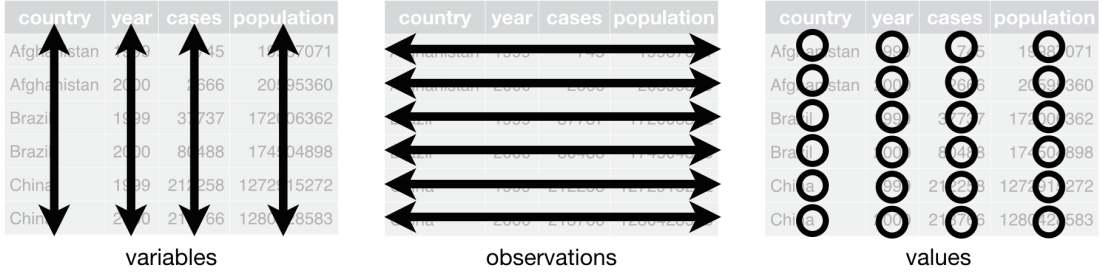
Magno Severino
Programa Avançado em Data Science e Decisão
30/10/2021

Introdução

- Uma mesma base de dados pode ser armazenada/representada em diferentes maneiras.
- Além disso, os formatos e convenções de armazenamento adotados podem variar muito, podendo dificultar análises.
- O processo de limpeza e organização de dados é praticamente obrigatório em qualquer contexto de análise de dados, e isso não se deve necessariamente a erros ou falhas no processo de armazenamento.

Tidy data

- Dados organizados em formato *tidy* são facilmente usados para análise e modelagem.
- Regras que definem um conjunto de dados *tidy* [1]:
  - Cada variável deve ter sua própria coluna;
  - Cada observação deve ter sua própria linha;
  - Cada valor deve ter a sua própria célula.



- O formato de uma tabela *tidy* depende do que são considerados **variável** e **observação** em um dado contexto.

[1] Wickham, H. Tidy data. Journal of statistical software, v. 59, n. 1, p. 1-23, 2014.
[2] Figura do livro R for Data Science Wickham, H. and Grolemund, G. 2017.

“*Tidy datasets are all alike, but every messy dataset is messy in its own way*.”
Hadley Wickham

Violações das regras *tidy* mais comuns

Erro	Solução	Função no R
Uma variável separada em várias colunas	Transformar várias colunas em uma única e empilhar linhas	pivot_longer
Variáveis diferentes empilhadas como linhas	Transformar as linhas repetidas em colunas	pivot_wider
Informação espalhada em várias tabelas	Mesclar as tabelas pelo valor de uma coluna chave	***_join

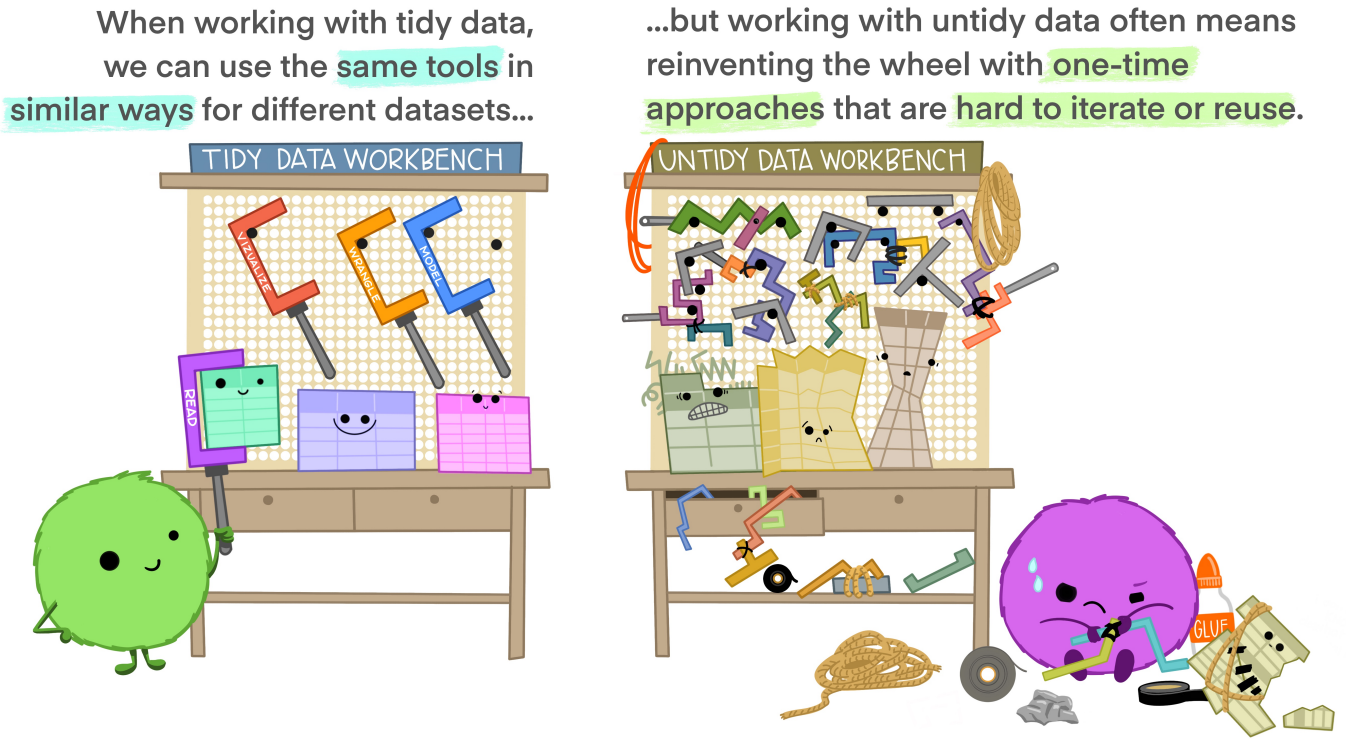


Figura de Allison Horst.

Tidy data

- Veja os dados de casos de tuberculose em diferentes paises.

```
library(tidyr)
table1
```

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

- Veja como é simples calcular a taxa de casos por população:

```
table1 %>%
  mutate(rate = cases / population * 10000)
```

Transformar em formato tidy

```
table4a
```

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

O que devemos fazer para deixar essa tabela no formato tidy?

Figura do livro R for Data Science Wickham, H. and Grolemund, G. 2017.

Pivot longer

```
table4a %>%
  pivot_longer(cols = -country,
               names_to = "year",
               values_to = "cases")
```

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

Transformar em formato tidy

```
table2
```

country	year	type	count
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

O que devemos fazer para deixar essa tabela no formato tidy?

Pivot wider

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

Figura do livro R for Data Science Wickham, H. and Grolemund, G. 2017.

Pivot wider

```
table2 %>%
    pivot_wider(names_from = type,
                values_from = count)
```

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Estudo de caso 1

O conjunto de dados `tidy::who` contém casos de tuberculose por ano, país, idade, gênero e método de diagnóstico fornecidos pelo *2014 World Health Organization Global Tuberculosis Report*.

```
library(tidyr)

View(who)
```

O pacote `janitor`

O pacote `janitor` disponibiliza algumas funções úteis para limpeza de bases de dados.

```
library(janitor)

library(tidyverse)
```

Algumas das funções mais úteis do pacote `janitor`:

- `clean_names()`: limpa os nomes das variáveis;
- `get_dupes()`: identifica linhas duplicadas;
- `remove_empty()`: remove qualquer coluna e/ou linha completamente vazia;
- `tabyl()`: constroi tabela de frequência.

Estudo de caso 2

Vamos trabalhar com os dados da planilha `pacientes.xlsx` [1].

```
library(janitor)

library(tidyverse)

library(readxl)

dados_brutos <- read_excel("pacientes.xlsx")

glimpse(dados_brutos)
```

[1] Dados do livro Ciência de Dados em R. Daminiani, A. et al., 2021.

Estudo de caso 3

Vamos trabalhar com dados públicos da ANFAVEA [1] e do IBGE [2].

- Série de autoveículos no arquivo `SeriesTemporais_Autoveiculos.xlsm`,
- Indice de volume e de receita nominal de vendas no comercio varejista no arquivo `tabela3416.csv`,
- Índice de receita nominal de serviços no arquivo `tabela6443.xlsx`,
- Série histórica do IPCA no arquivo `tabela1737.xlsx`.

[1] Página ANFAVEA.

[2] Página IBGE.

Obrigado!

magnotfs@insper.edu.br